IN THE CLAIMS:

~~1. A method of managing digital resources in a digital system, said method comprising the steps of:~~

~~reserving token values for certain digital resources in said digital system;~~

~~matching a selected token value in a free-buffer-queue to an incoming digital resource request;~~

~~moving said selected token value to a valid-request-queue;~~

~~removing said selected token value from said valid-request-queue to allow a digital agent in said digital system to process said incoming digital resource request; and~~

~~returning said selected token to said free-buffer-queue.~~

sub A2

2. The method of claim 1wherein said moving step includes the step of moving a packet of data associated with said incoming digital resource request into a memory location corresponding to said selected token value.

3. The method of claim 2 wherein said moving step includes the step of moving said incoming digital resource request from said memory location to allow said digital agent to process said incoming digital resource request.

sub A3

4. The method of claim 1 further comprising the step of blocking an incoming digital resource request when said free-buffer-queue is empty.

5. The method of claim 1 further comprising the step of preferentially removing said selected token value from a priority valid-request-queue.

5 ~~6~~. A digital system, comprising:

(A) a transmission channel to route an incoming digital resource request;

(B) a digital agent to process said incoming digital resource request; and

(C) a token controller positioned between said transmission channel and said digital agent, said token controller including

(1) a free-buffer-queue,

16

(2) a valid-request-queue,

(3) a memory,

(4) a token-based request processor connected to said transmission channel, said free-buffer-queue, said valid-request-queue, and said memory, said token-based request processor being configured to

(a) match said incoming digital resource request with a selected token value in said free-buffer-queue,

(b) move said incoming digital resource request into said memory, and

(c) place said selected token value in said valid-request-queue, and

(5) a token-based responder connected to said free-buffer-queue, said valid-request-queue, said memory, and said digital resource, said token-based responder being configured to

(a) move said selected token value from said valid-request-queue into said free-buffer-queue, and

(b) route said digital resource request from said memory to said digital agent.

6 ~~7.~~ The apparatus of claim 5 ~~6~~ wherein said token-based request processor is configured to block an incoming digital resource request when said free-buffer-queue is empty.

7 ~~8.~~ The apparatus of claim 5 ~~6~~ wherein said token-based request processor selectively places high priority incoming digital resource requests into a priority valid-request-queue of a set of priority valid-request-queues, and said token-based responder preferentially removes tokens from said priority valid-request-queue.

8 ~~9.~~ The apparatus of claim 5 ~~6~~ wherein said free-buffer-queue is constructed as a hardware-based First-In-First-Out device.

9 ~~10.~~ The apparatus of claim 5 ~~6~~ wherein said valid-request-queue is constructed as a hardware-based First-In-First-Out device.

17

10 ~~11~~. The apparatus of claim 5 ~~6~~ wherein said free-buffer-queue is constructed as a software-based write-only FIFO using a write index cache.

11 ~~12~~. The apparatus of claim 5 ~~6~~ wherein said valid-request-queue is constructed as a software-based write-only FIFO using a write index cache.

12 ~~13~~. A digital system, comprising:

(A) a transmission channel;

(B) a plurality of digital agents; and

(C) a plurality of token controllers, each of said token controllers being positioned between said transmission channel and a selected digital agent of said plurality of digital agents, each of said token controllers including

(1) a free-buffer-queue,

(2) a valid-request-queue,

(3) a multi-port memory,

(4) a token-based request processor connected to said selected digital agent, said free-buffer-queue, and said multi-port memory, said token-based request processor being configured to

(a) match an incoming digital resource request from said selected digital agent with a selected token value in its free-buffer-queue,

(b) route said selected token value to a valid-request-queue of a destination token controller of said plurality of token controllers, and

(c) move said incoming digital resource request into said multi-port memory, and

(5) a token-based responder connected to said valid-request-queue and said transmission channel, and said selected digital agent, said token-based responder being configured to

(a) read a specific token value from its valid-request-queue, said specific token value originating from a digital resource request generated by a source token controller,

(b) receive said digital resource request from a multi-port memory of said source token controller, and

(c) pass said digital resource request to said selected digital agent associated with said token-based responder.

13 ~~14.~~ The apparatus of claim 12 ~~13~~ wherein said token-based request processor includes a router to route said selected token value to said valid-request-queue of said destination token controller.

14 ~~15.~~ The apparatus of claim 12 ~~13~~ wherein said token-based responder is configured to route said specific token value to a free-buffer-queue of a source token controller.

~~16. A method of managing digital resources in a digital system, said method comprising the steps of:~~

~~reserving token values for certain digital resources in said digital system;~~

~~matching a selected token value in a free-buffer-queue to an incoming digital resource request;~~

~~moving said selected token value and said incoming digital resource request to a digital agent for processing; and~~

~~returning said selected token to said free-buffer-queue.~~

~~17. The method of claim 16 further comprising the step of performing said reserving, said matching, and said moving steps with a first processor and executing said returning step with a second processor.~~

19